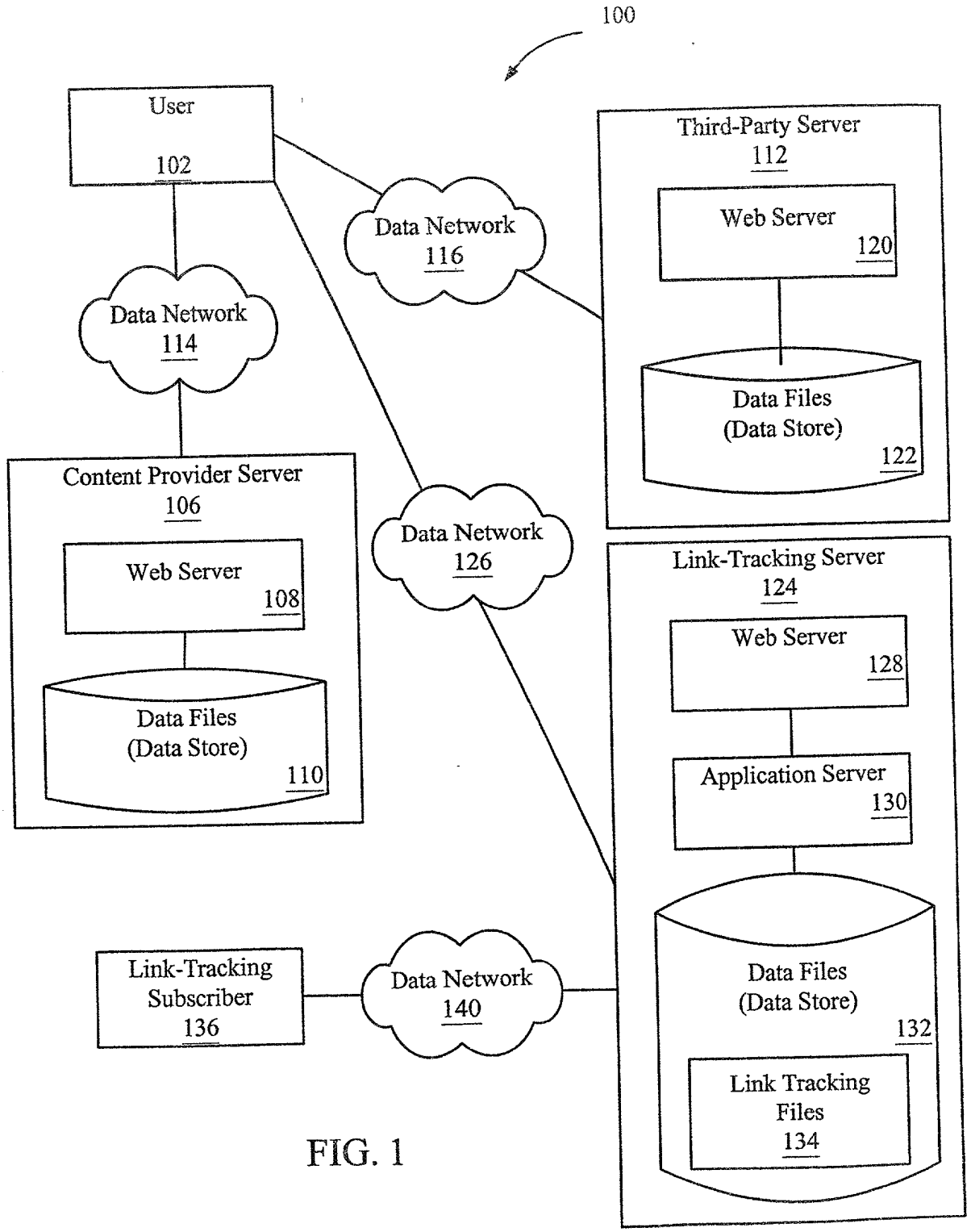
100
User
102
Data Network
114
Content Provider Server
106
Web Server
108
Data Files
(Data Store)
110
Data Network
116
Third-Party Server
112
Web Server
120
Data Files
(Data Store)
122
Data Network
126
Link-Tracking Server
124
Web Server
128
Application Server
130
Data Files
(Data Store)
132
Link Tracking
Files
134
Link-Tracking
Subscriber
136
Data Network
140

FIG. 1

FIG. 2A

```
<!-- BEGIN Event Tracking Code -->
<script language="javascript1.2">
function recordit(k, w)
//This function records the link.  Parameters k and w are passed.  K is the link and w is an array of stored
functions.
        {
        var gettext="NA";
        //This initializes the variable used to hold the text that the link wrapped.
        var iloc=0;
        while (document.links[iloc]!=k){
                iloc++;
                }
        //This is a loop which determines which element in the array the link that was passed.
        if (w[iloc]){w[iloc]();}
        //This checks to see if a function was stored for that link.  If so, it executes the stored function.
        document.all?gettext=k.innerHTML:gettext=k.text;
        //This sets the gettext variable to the text or html that was wrapped by the link.
        document.images.hitboximage.src="http://ww3.hitbox.com/logs/log.cgi?link="+k.href+"&iloc="+iloc+"
&named="+gettext;
        //This sends the link information back to the hitbox servers.
        }

```

FIG. 2B

```
function initial()
//This function sets the onclick event for each link and stores prior events associated with it, if any.
        {
        var storef = Array(document.links.length);
        //This array will store prior functions.
        for (var i = 0; i < document.links.length; i++) {
        //This loop runs through all the documents on a page.
                if (document.links[i].onclick) storef[i]=document.links[i].onclick;}
                //If a function is associated with that link it is stored in an array.
                        document.links[i].onclick = function (){
                        recordit(this, storef);
                        return true;
                        //Here is where the onclick event for each link is associated with the recordit function.
                        }
                }
        }

window.onload = initial;
//This starts the process when the window has loaded.
</script>

<IMG SRC="http://stats.hitbox.com/buttons/ct0.gif" HEIGHT=68 WIDTH=82 BORDER=0
NAME="hitboximage">
<!-- This is the initial image used to record a page view.
```

FIG. 3

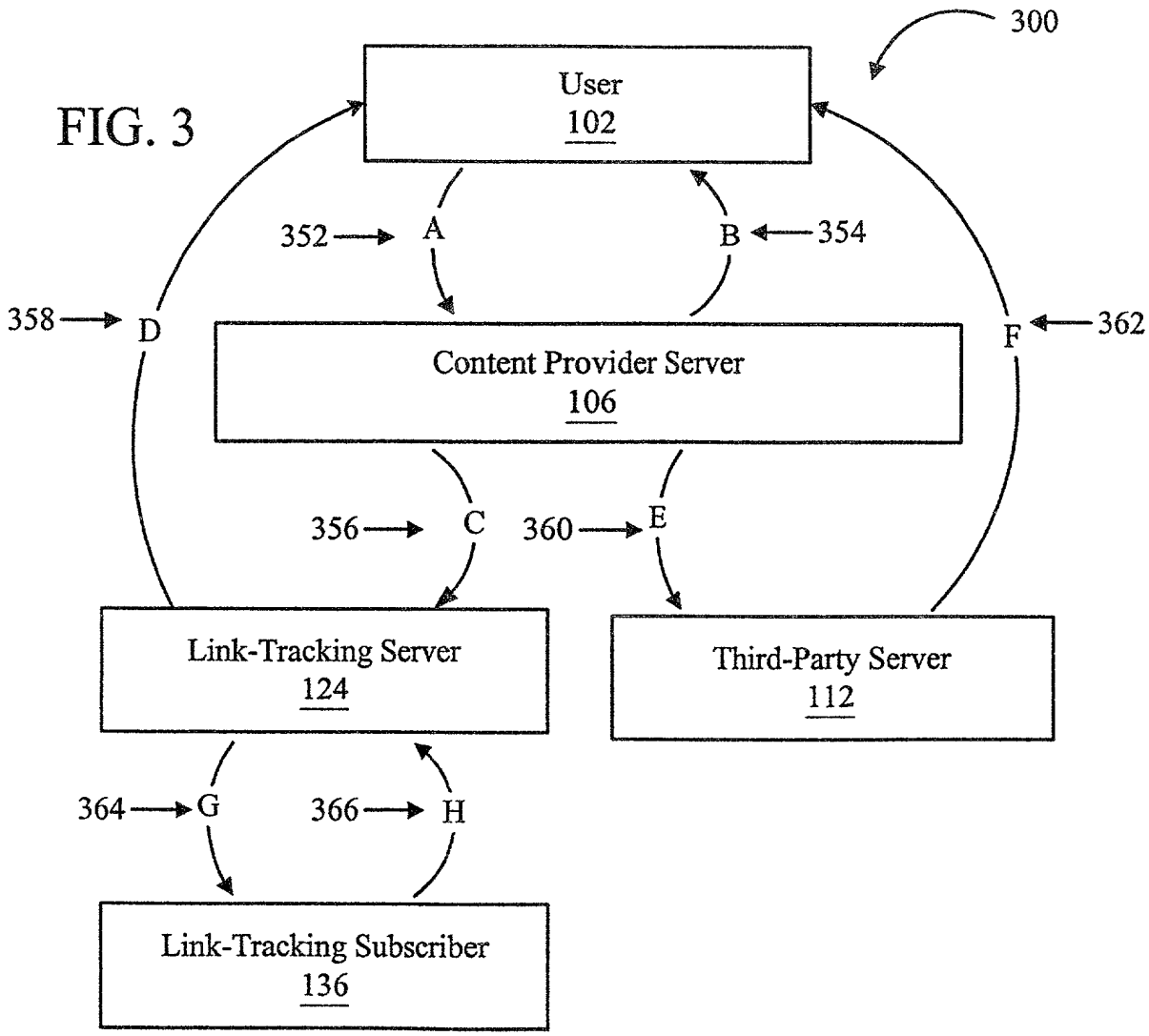
300
User
102
352
A
B
354
358
D
F
362
Content Provider Server
106
356
C
360
E
Link-Tracking Server
124
Third-Party Server
112
364
G
366
H
Link-Tracking Subscriber
136

FIG. 4A

```
<script language="javascript1.2">
This code demonstrates how to record downloads and exit links using the invention.
_hbi=new Image();
_ot=0;
//Initializes variables.

function _gf(_x){
//This function retrieves the file or document name requested in a URL.
    _ed = (_x.length);
    _be = ((_x.lastIndexOf("/"))+1);
    _fn = (_x.substring(_be, _ed));
    return _fn;
    }

function _sb(_k){
//This function checks to see if the link is an exit link.
    eval("var _gt=/(\\w*\\.)*(\\w*\\.)/i;");
    _gt.exec(location.hostname);
    //This parses the link for domain name.
    var _fr="co.com.org.net.";
    if(_fr.indexOf(RegExp.$2)!=-1){
            eval("var _uf=/(\\w*\\.)*(\\w*\\.)(\\w*\\.)/;");
            _uf.exec(location.hostname);}
    //This is additional checking in case the link is to a foreign domain.
    if(_k.href.indexOf(RegExp.$2)==-1){_hbi.src="http://ehg.hitbox.com/HG?el="+escape(_k.href);}
    //If the domain name is not the same as the current document, the information is sent back to the hitbox
servers.
    return true;
    }
```

FIG. 4B

```
function _ri(_k,_w){
//This function recieves the link and checks if it is a download. Parameters _k and _w are passed. _k is the
link and _w is an array of stored functions.
    var _ilc=0;
    while(document.links[_ilc]!=_k){_ilc++;}
    if(_w[_ilc]){_w[_ilc]();}
    //This is a loop which determines which element in the array the link that was passed.
    _fl=_gf(_k.pathname);
    //Sets variable _fl to the file or document name in the link.
    if(_fl!=""){
    //Checks if name exists
            var _pt=new RegExp("\..?html?|\.asp|\.cfm|\.jsp|\.cgi|\.php[3-4]?|\.pl|\.taf|\.dll","i");
            if(!_pt.test(_fl)){
            //Checks to see if file extension is a among a list of known HTML-types.
                    _hbi.src="http://ehg.hitbox.com/HG?fn="+escape(_fl);}
            //If the file is not among that list, the file information is sent back to the hitbox servers.
            else{
                    _sb(_k);}}
            //If info was not sent back to hitbox server, it checks for exit links.
    else{
            _sb(_k);
            //If info was not sent back to hitbox server, it checks for exit links.
            }
    return true;
    }
```

FIG. 4C

```
function _it(){
//This function sets the onclick event for each link and stores prior events associated with it, if any.
    if(_ot==1)_wo();
    //If an onload event was priviously set, executes it now.
    var _sf=new Array(document.links.length);
    //This array will store prior functions.
    for(var i=0;i<document.links.length;i++){
            _rv="true";
            if(document.links[i].onclick){
            //This loop runs through all the documents on a page.
                    _sf[i]=document.links[i].onclick;
                    //If a function is associated with that link it is stored in an array.
                    if(_sf[i].toString().indexOf("return false")!=-1)_rv="false";
                    //If the return value of the stored function is false, the return value for the new function is
set to false.
                    }
            eval("document.links[i].onclick=function(){_ri(this,_sf);return "+_rv+";}");}
            //Here is where the onclick event for each link is associated with the _ri function.
    }

    if(window.onload){_wo=window.onload;_ot=1;};window.onload=_it;
    //If a function is already associated with the window loading, it is stored and executed later.
</script>
```